**Scheduling Tool Documentation**

The objective of this file is to culminate all relevant information related to the usage and dunderstanding of the Scheduling Tool software. Because this application was built for the UVA Blacksmithing club, I have not spent time to improve the user interface. As a result, this documentation will provide a helpful tutorial for first time users, as the software does not have exception handling and other features which might otherwise help users troubleshoot their issues. It is my goal to accurately and approachably explain this process so that anyone-no matter their background-can successfully use the program to create a club schedule.

Table of Contents

**Overview2**

Features2

Background2

Penalty Term2

Supported Schedule Layouts3

Output4

Limitations & Support4

**Preparation5**

Setup5

Inputs5

File Location5

Layout Selection5

Member Data6

**Software Usage**7

Runtime7

User Interface8

1. Enter the file location8
2. Enter the file name8
3. Drop the last row9
4. How many weeks10
5. How many sessions10
6. How many alternating10
7. Day x (every week)10
8. Day x (every other week)11
9. Where to save files 11
10. Optional Identifier 11
11. Preview & Completed12

Troubleshooting13

**Features**

**Overview**

**Background**

As a result of high club interest, it became necessary for the UVA Blacksmithing Club expand the number of weekly sessions available to its members to satisfy demand. Before this change, forging sessions were limitted to Friday 1-5pm each week, and anyone not available for this time was simply not able to participate in the sessions.

The Scheduling Tool was built to create weekly rosters (of 2 or 3 forging sessions per week) for the UVA Blacksmithing Club that evenly distribute forging opportunities among club members for a given semester. It is important to note, though, that it makes certain assumtions about the layout and priorities of the schedule.

Assumptions:

1. There is only room for 6 club members during each forging session
2. Because this is a college club, there are a limitted number of session times that are possible for both club members and the forge.
   1. These times are: Monday through Friday from 1-5pm, with an additional session time available from 9am-1pm on Friday. Friday morning sessions are denoted “FriM,” or, “friM,” and Friday afternoon sessions are denoted “FriA,” or, “friA,” respectively.

Priorities:

1. Every club member is assigned 1 session
2. For session times where 6 or fewer members are available, those available members are automatically scheduled for this time.
3. Every club member will be scheduled for biweekly sessions if they are not already
4. For a given session that is not yet filled, for which more than 6 club members are available; club members will be scheduled for these sessions based on the total number of sessions they are already scheduled for. This ensures that club members have equal access to weekly forging sessions.

**Penalty Term**

The system of scheduling described thus far relies on the club members providing enough availability to forge on multiple days of the week. Without that, the club cannot increase the amount of forging sessions available, and the club gets gridlocked into dividing the one session per week among 30 or more club members.

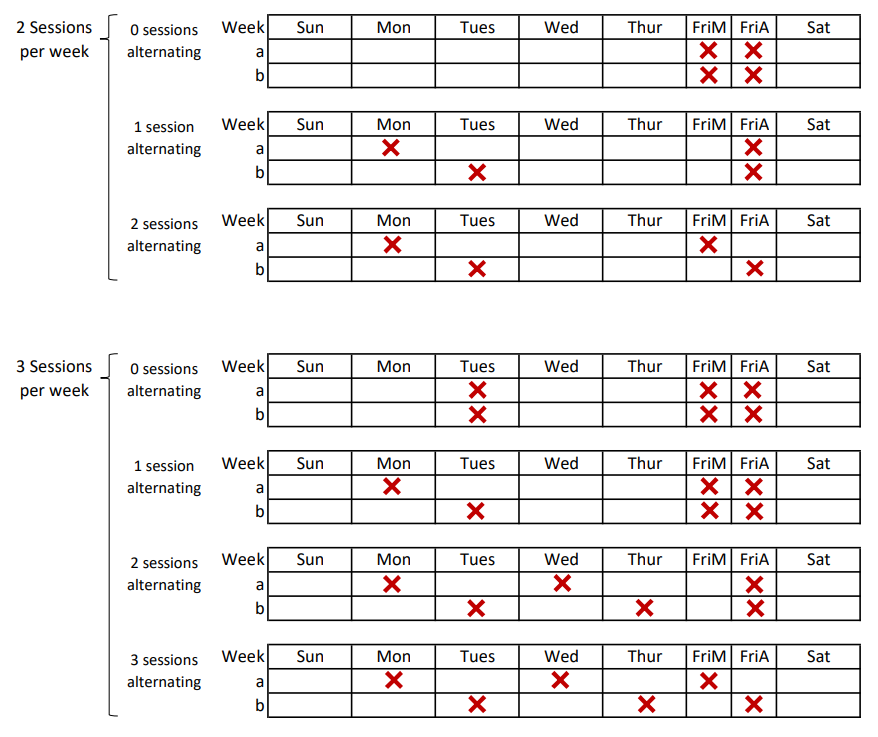
Club members who are available for less common times are, sadly, penalized for it by being automatically scheduled for all those sessions. The implication of this system is that club members who are available on Monday and Friday afternoon will have most of their forging sessions on Monday, which is the less desirable time of the two. Because of this, it is unfair to these club members if other members do not provide their full availability in order to aviod being scheduled for a less desirable time.

The penalty term is a feature of the Scheduling Tool which allows the user to give a club member a reduced amount of scheduled sessions. This is the only exception to the priorities listed in the prior secion. It will schedule such a member for every third week of the semester. This penalty is intended to be used very sparingly, if at all, and its main function is to enable actual reprocussions if club members are unwilling to cooperate-to the detriment of the club.

**Supported Schedule Layouts**

The introduction of multiple weekly sessions allows for a wide variety of schedule designs. Even with 1 weekly session, it would be possible to alternate the session time between the two times most available to club members. Given 2 sessions per week, this proposition becomes more complex. I have included all possible session layouts within this program, so it is possible to create the optimal schedule for your semester’s availablility. I have also included an option to create a schedule with 3 sessions per week to account for further club growth.

Below is a diagram illustrating all possible schedule layouts that can be created using this software.



**Output**

Two files are created from the Scheduling Tool software.

One is a .csv file which contains only the club schedule. This file is meant to be distributed using the club listserv, and in conjunction with the document “Personal Schedule How-To,” so that club members can see their individual schedules for forging sessions.

The other file is also a .csv, but it contains summary statistics for the club schedule. This file allows the user to see how even the sessions are distributed, and see the number of sessions scheduled for each club member. This information could be useful if saved over time, as it would allow future club leaders to see the number of sessions club members get per semester, on average. With a couple excel formulas you can make a histogram of the results.

Example output:

Summary Statistics Club Schedule

Graphical user interface, application, table, Excel

Description automatically generatedGraphical user interface, text, application, table

Description automatically generated

Note: There are blank spots in this schedule, indicating open forging session slots that cannot be filled using the current schduled days. Jane doe has less sessions scheduled than everyone else. These are indications that the schedule is not optimized.

**Limitations & Support**

The capability of this software is limitted by the assumptions explained in the “Background” subsection of “Features” (Page \_\_). Beyond the program’s functional limitations, the user may experience compatibility issues with the application and their operating system. If you are experiencing compatibility issues you are welcome to visit my Github page in order to download the source code and find a solution locally.

Github link:

**Preparation**

**Setup**

There setup of the Scheduling Tool is very simple. There are two versions of the software:

* Scheduling Tool Win (Windows compatible)
* Scheduling Tool Mac (Mac OS X compatible)

Make sure to download the version that is compatible with the operating system at your disposal. If neither of these operating systems prove accessible, refer to the Github link in the “Limitation & Support” section, and install a version of python (3.8+) that is compatible with your operating system. At this point, you may run the Scheduling Tool source code directly from python.

**Inputs**

**File Locations**

The location of the main input file, and the location where you wish to save the final documents, are both required by the Scheduling Tool software. This is called a “path.”

**Layout Selection**

Before you begin, use the diagram in the “Supported Schedule Layouts” section to determine your desired number of:

* Sessions per week
* Sessions that will alternate

These numbers are required inputs in the Scheduling Tool software.

**Member Data**

In order to use the Scheduling Tool software, you must have a prepared csv file which specifies each club member, their respective availability, and their penalty term. The prepared file should mirror the snippet below, which was prepared using fake data to mimic a realistic availability pattern. The column “Availability” is not necessary; it is a hold-over from prior manipulations.

Note:

1. The file MUST be a .csv
   1. In order to save an excel file as a csv,
2. The column labels must exactly match those in the table below

Table

Description automatically generated with low confidence

**Software Usage**

**Runtime**

Because python is not suited for applications run through executable files, the Scheduling Tool is quite slow to run. When running the application for the first time, or after restarting your computer, the application will take longer to run. I am making a particular note of this because the executable can take up to 10 minutes to run at the extreme. Once it has run once, it will take around 1 minute to 30 seconds to run subsequently.

What do I mean by “to run?” The code I wrote for this software must be compiled and accessed by the packaging software that turned my code into an executable file. While this is happening, a window will appear with a blinking cursor, an example of which is shown below.

A computer screen capture

Description automatically generated with medium confidence

Once you see text in the window, the wait is over, and the application runs very quickly whilst building the schedule.

**User Interface**

The user interface for this software is very basic. This section will guide the user in properly inputting required data in order to run the Scheduling Tool successfully.

1. Enter the file location:

In this field, copy and paste the path of the the .csv file that contains the relevant member data. The formatting of this .csv file is covered in depth on page 8 of this document.

Shape, rectangle

Description automatically generated with medium confidence

Once you do so, you must add a backslash to the end of the path, as this backslash is not included in the path copied from the file’s properties.

This is the correct format:

Text

Description automatically generated

1. Enter the file name:

This command is looking for the name of the .csv file tha contains relevant member data. Be sure to include the format suffix in your input!

Text

Description automatically generated

1. Drop the last row?

Preceding this input box is an output of the table as the software reads it. This allows the user to verify that the correct information has been accessed. In some instances, the last row of the data will be filled with NaN values. In this case, the user will enter “yes” in the input field. Any other input will result in the last row being retained.

A computer screen capture

Description automatically generated with medium confidence

Following this input, the software will output the data again to allow the user to again verify that the data to be used in the software is correct.

A computer screen capture

Description automatically generated with medium confidence

1. How many weeks?

Simply input the total number of weeks the desired schedule should include. Note that you must account for spring and fall break, and forging sessions ususally end on the last week of classes.

Text

Description automatically generated with medium confidence

1. How many sessions?

Input the number of forging sessions that will occur during each scheduled week.

Text

Description automatically generated

1. How many alternating?

Input the number of forging sessions whose scheduled time is not the same each week.

Text

Description automatically generated

1. Day x (every week):

This field will occur if the number of alternating session times is less than the number of weekly sessions. The entered session time, in this case Friday afternoon (1-5pm), will be on the schedule every week.

Text

Description automatically generated

1. Day x (every other week):

This field will occur if the number of alternating session times is greater than zero. This input is structured to follow the club resources, not the session time. Think of this as alternating the session time around for the “Day” the club has funding for. Thus, “option 1,” and, “option 2,” are the times that will alternate with one another for a given two week cycle. The example below exactly mirrors the diagram for “2 sessions per week, 1 session alternating,” on page 4 of this document.

Text

Description automatically generated

1. Where to save files?

This input field is asking for the location where the user wants to have the output files created. To place them in the same folder as the input .csv file, simply paste the same path into this field. Remember to add the backslash just as in step 1!

Text

Description automatically generated

1. Optional Identifier:

This field is intended to help users organize and meaningfully label club schedules. The inputted string will follow both the base names “new\_stats” and “new\_schedule” for the schedule statistics and schedule outputs respectively.

Text

Description automatically generated

1. Preview & Completed

You’re done!

The software creates a preview of the two output files, and the user is free to press enter to close the program whenever they wish.

Example outputs are shown on page 5 of this document. Note that those outputs correspond to the inputs demonstrated in this section.

A computer screen capture

Description automatically generated with medium confidence

Here is a snippet of my folder “Sample Club Folder”. Both of the output files were saved in the same location as the original members data .csv file.

Graphical user interface, text, application, email

Description automatically generated

**Troubleshooting**

If there is any error in your inputs, the software will error out and close immediately upon the user entering that data. This will help you find your mistakes in inputs.

Things to look for:

* Typos
  + This is by far the largest cause of errors. It is very easy to forget an underscore or a capitalized letter, and small errors like these will cause the Scheduling Tool software to error.
* Slash orientation
  + If you have checked all your input thoroughly for typographical errors, and you are running into trouble while loading your members data .csv into the software or outputting the files from the software, there is one more technique you can try. Copy the desired path as usual, but change all the backslashes (\) to forward slashes (/), subsequently entering this new path into the software. This is a frequent culprit of errors.